Computer Programs for Moving Data Between Databases in the U.S. Geological Survey's National Water Information System

by Gary Rogers
(grogers, (406) 441-1319)

1. Introduction

This document describes wat_move, a system of computer programs developed for moving National Water Information System (NWIS) data between the U. S. Geological Survey's distributed computer databases. The system was originally documented in U. S. Geological Survey Open-File Report 91-241. Since that report was written, the software has been ported from the PRIMOS-based minicomputer environment to the UNIX-based workstation and file server environment. Use of the UNIX version is somewhat different from the PRIMOS version; therefore, new documentation is appropriate. This document provides sufficient information for operating wat_move. However, Open-File Report 91-241 is still useful for more detailed information.

The wat_move program works with files for the ADAPS, GWSI, and QW subsystems of NWIS and consists of three subsystems: one for retrieval, one for loading, and one for purging. The retrieval subsystem (wat_get) creates a control file and transaction files of retrieved data for transfer and initiates a file transfer to send the transaction files to a receiving site. The loading subsystem (wat_put) reads the control and transaction files retrieved from the source database and loads the data into appropriate files. The purging subsystem (wat_purge) deletes data from a database.

This document provides information on the following topics:

- differences from the Prime version
- general operation
- operation of the wat_get subsystem
- operation of the wat_put subsystem
- operation of the wat_purge subsystem
- format of the file used for selecting sites
- format of the file used for selecting NWIS files
- description of an option for editing NWIS files during processing
- discussion of cron processing
- qwindex file

2. Differences from the PRIMOS Version

The UNIX version of wat_move should be easier to use than the PRIMOS version. The following are the major differences between the UNIX version and the PRIMOS version:

- The wat_move program no longer needs to be separately installed as it is installed as part of the NWIS software.
- Configuration files are now used to control processing. These files replace the user-modified portions of the Command Procedure Language (CPL) programs used on the Prime.
- Only the NWIS files actually required for processing will be opened.
- Retrieval or deletion of ADAPS Unit-Values data may be done for a specific range of dates, rather than just a range of water years.
- The user-coded routines mod_xtr, mod_bld, and tst_del may no longer be used. The functionality of these routines is provided by user-coded edit commands (sec. 9).

3. General Operation

The wat_move program is invoked by entering either:

```
wat_move
```

or

```
wat_move <subsystem>
```

where <subsystem> is “wat_get”, “wat_put”, or “wat_purge”.

If the program is invoked without a subsystem argument, the wat_move menu is displayed:

```
------------------------------------------------------------------------------
wat_move  Rev NWIS_1.1_+19970511

Initiated: 970708.173346

wat_move options

 1 -- wat_get, retrieve [and transfer] NWIS data

 2 -- wat_put, store data retrieved using wat_get

 3 -- wat_purge, delete NWIS data


 9 -- Exit to Operating System

Please enter a number from the above list or a Unix command:
------------------------------------------------------------------------------
```

The user can execute one of the subsystems, a UNIX command, or return to the operating system.

If the program is invoked with a subsystem argument that subsystem is executed without user interaction and no log file will be produced. (Use output re-direction or “tee” command to capture output). This mode would normally be used for background or cron processing. See section 10 for more information about cron processing.

Sections 4-6 provide detailed information about the set-up required for each of the subsystems.

4. Operation of the wat_get Subsystem

The wat_get subsystem retrieves data from an NWIS database and, optionally, transfers that data to another Internet node using the File Transfer Protocol (FTP). An option of the wat_get subsystem is to count records only which could be used to estimate disk space requirements. Operation of the program is described in the following sections (4.1-4.3).

4.1 Invoking wat_get

The program may be run from any directory and is invoked from the wat_move main menu or by entering “wat_move wat_get”. Operation of the program is controlled by three files:

1) a configuration file, wat_get.cfg (section 4.2)

2) a “site-list” file listing the sites to be retrieved (section 7)

3) a “file-list” file defining the NWIS files to be accessed (section 8)

4.2 wat_get Configuration File, wat_get.cfg.

The wat_get.cfg file defines a number of parameters required for operation of wat_get. A prototype of the wat_get.cfg file can be retrieved from $NWISHOME/data/auxdata/wat_move/wat_get.cfg.master. This file is internally documented and contains a section to be edited by the user. The following is an example of a wat_get.cfg file:

```
#! /bin/sh
#
#  wat_get.cfg
#
#  Configuration file for wat_get program
#
#  Notes:  The variable names must be in upper-case.
#          No space is allowed on either side of the equals sign.
#
#  The variables are defined as follows:
#    DIS_NODE        :  DIS node ident (e.g., qvarsa)
#    DBNUM           :  Database number for retrieval
#    START_DATE      :  Start date (yyyymmdd) for ADAPS UV retrieval,
#                       NULL (‘’) => earliest date available
#    END_DATE        :  End date for UV retrieval, NULL => current date
#    SITE_LIST_PATH  :  Path name of site-list file
#    FILE_LIST_PATH  :  Path name of file-list file
#    EDIT_LIST_PATH  :  Path name of edit-list file, NULL if not used
#    XFR_DIR         :  Path name of transfer directory.
#    MODE           :  Processing mode, W to write tran files, C to count only
#    FTP_DEST        :  IP address of destination machine for ftp transfer
#    FTP_USER        :  User ID on destination machine
#    FTP_PW          :  User password on destination machine
#    FTP_PATH        :  Pathname of directory on destination machine
#
#  Modify the following lines as necessary:
#-----------------------------------------------------------------------------
```

```
DIS_NODE=qqqqqq
DBNUM=01
START_DATE=''
END_DATE=''
SITE_LIST_PATH=./site_list
FILE_LIST_PATH=./file_list
EDIT_LIST_PATH=''
XFR_DIR=./data
MODE=W
FTP_DEST=XXXXXX
FTP_USER=anonymous
FTP_PW=wat_move
FTP_PATH=incoming/wat_move/data
```

A more detailed description of each of the variables follows:

DIS_NODE : A 6-character label for the source node. This should be the last 6 characters of the domain name, e.g. "qvarsa".

DBNUM : 2-character database number.

START_DATE : Start date for retrieval of ADAPS unit-values data. The date is in the form yyyymmdd. If a null ('') start date is specified the data will be retrieved from the earliest date available.

END_DATE : End date for ADAPS unit-values retrievals. If null, retrieval will continue through the latest date available.

SITE_LIST_PATH : Path name for site-list file (section 7.0).

FILE_LIST_PATH : Path name for file-list file (section 8.0).

EDIT_LIST_PATH : Path name for edit-list file if user-defined editing is to be done (section 9.0).

XFR_DIR : Path name of a directory for temporary storage of the retrieved data. This directory may require a lot of free disk space. See variable "MODE" for a way to estimate the space required. Note that any files existing in this directory when wat_get is invoked will be deleted.

MODE : Processing mode, either "W" for write or "C" for count. If "W" is used, data will be retrieved and written to XFR_DIR. If "C" is used, the retrieved data will not be written but only counted with an estimate of the disk space required.

FTP_DEST : Internet node address of destination machine for the FTP transfer of the retrieved data. If this value is "XXXXXX" no transfer will be attempted.

FTP_USER : User id on the destination machine.

FTP_PW : User's password on the destination machine.

FTP_PATH : Pathname of the directory to which the data are to be transferred.

4.3 Output from wat_get

There are two files output by wat_get that may be transferred to another site:

- A control file named qqqqqq.ctrl where qqqqqq is the "DIS_NODE" defined in the wat_get.cfg file. This file lists the date-time of the retrieval as well as information about each of the NWIS files accessed. This file consists of a record indicating the date-time of the retrieval in the form:
  yyyymmddhhmm
  and one or more records indicating the NWIS file identification (see sec. 7), the name of the transaction file, and the number of records retrieved. The format of these records is:
  AAAA FFFFFFFFFFF NNNNN
  where AAAA (cols. 1-4) is the NWIS file identification, FFFFFFFFFFF (cols. 6-16) is the file name, and NNNN (cols 18-22) is the number of records.
- One or more transaction files named qqqqqq.ffff where qqqqqq is the "DIS_NODE" and "ffff" is the NWIS file identification (see sec. 7).

If ftp transfer is specified the files will be compressed before transfer (gzip compression).

If the program was invoked using the wat_move main menu, a log file named wat_get.date.time.log will also be created. This file lists the NWIS files accessed with number of records retrieved, as well as any error messages. This information will be provided to the user by e-mail if a cron entry is used to invoke the program (see sec. 3) The following is a list in a typical wat_get log file:

```
------------------------------------------------------------------------------
wat_get initiated: Tue Jul 8 16:17:40 EDT 1997

Processing in Ingres database: nwiqqvarsa::nwisdb
Deleting all files in transfer directory ./data ...

watget rev NWIS_1.1 -- Extract NWIS data for transfer -- 19970708 1617

Enter node ID: qqqqqq
Enter database number (<CR>=1): 01
Enter path name of site-list file (<CR>=site_list): ./site_list
Enter path name of file-list file (<CR>=file_list): ./file_list
Enter path name of transfer directory (<CR>=./data): ./data
Enter start date for ADAPS UV file retrieval (yyyymmdd, <CR>=epoch):
Enter end date (<CR>=current):
Enter  W  to write files,  C  to count records only: W

The following buffer editing commands have been specified--
set (1:5) = ‘WTMV ‘ if nwisid != ‘QW  ‘
set (9:13) = ‘WTMV’ if nwisid = ‘QW  ‘

Extracting NWIS data ...

FILE  TOTAL  KILO-
 ID   TRANS  BYTES  FILE-DESCRIPTION
```

```
SITE      3      2  Site File
QW      800    515  QW File

Total kilobytes retrieved:    517
FTP user (DG/UX TCP/IP Release 5.4R3.10) ready.

wat_get finished: 970708.161933
------------------------------------------------------------------------------
```

5. Operation of the wat_put Subsystem

The wat_put subsystem loads NWIS data that were retrieved using wat_get. Operation of the wat_put subsystem is described in sections 4.1 - 4.3.

5.1 Invoking wat_put

The wat_put subsystem may be run from any directory and is invoked from the wat_move main menu or by entering “wat_move wat_put”. Operation of wat_put requires three files:

1) a configuration file, wat_put.cfg (section 5.2)

2) a “node list” file listing the nodes to be processed (section 5.3)

3) a “file-list” file listing the NWIS files to be processed (section 8)

5.2 wat_put Configuration File, wat_put.cfg.

The wat_put.cfg file defines a number of parameters required for operation of wat_put. A prototype of the wat_put.cfg file can be retrieved from $NWISHOME/data/auxdata/wat_move/wat_put.cfg.master. This file is internally documented and contains a section to be edited by the user. The following is an example of a wat_put.cfg file:

```
#! /bin/sh
#
#  wat_put.cfg
#
#  Configuration file for wat_put program
#
#  Notes:  The variable names must be in upper-case.
#          No space is allowed on either side of the equals sign.
#
#  The variables are defined as follows:
#    DBNUM           :  Database number for retrieval
#    NODE_LIST_PATH  :  Path name of node-list file
#    FILE_LIST_PATH  :  Path name of file-list file
#    EDIT_LIST_PATH  :  Path name of edit-list file, NULL (‘’) if not used
#    RCV_DIR         :  Path name of transfer directory
#
#  Modify the following lines as necessary:
#-----------------------------------------------------------------------------
DBNUM=01
NODE_LIST_PATH=./node_list
```

```
FILE_LIST_PATH=./file_list
EDIT_LIST_PATH=''
RCV_DIR=./data
```

A more detailed description of each of the variables follows:

DBNUM :	2-character database number, for example, 01.
NODE_LIST_PATH :	Path name for node-list file (section 5.3).
FILE_LIST_PATH :	Path name for file-list file (section 8)
EDIT_LIST_PATH :	Path name for edit-list file if user-defined editing is to be done (section 9).
RCV_DIR :	Path name of a directory which contains the control file(s) and transaction files retrieved using wat_get.

5.3 Description of the node-list File

The node-list file lists the nodes (see "DIS_NODE" in section 4.2) for which received data are to be processed by wat_put. This file must be created by the user using an editor. The node name (DIS_NODE) is in columns 1-6 of each record in the node-list file. The file will generally contain only one line that indicates the name of the node from which the data were retrieved. For example, if the data were retrieved from nwisdmthln.cr.usgs.gov, the node-list file would contain the line:

```
dmthln
```

5.4 Output from wat_put

If the program was invoked using the wat_move main menu, a log file named wat_put.date.time.log will be created in each run of wat_put. This file lists the NWIS files accessed with the number of records loaded, as well as any error messages. This information will be provided to the user by e-mail if a cron entry is used to invoke the program (see section 3). The format is similar to that of the log file produced from wat_get, described in section 4.3.

6. Operation of the wat_purge Subsystem

The wat_purge subsystem deletes data from an NWIS database. Operation of wat_purge is described in the following sections (6.1-6.3).

6.1 Invoking wat_purge

The wat_purge subsystem may be run from any directory and is invoked from the wat_move main menu or by entering "wat_move wat_purge". Operation of the program requires three files:

1) a configuration file, wat_purge.cfg (section 6.2)

2) a site-list file listing the sites for which data is to be deleted (section 7)

3) a file-list file defining the NWIS files to be processed (section 8).

6.2 wat_purge Configuration File, wat_purge.cfg.

The wat_purge.cfg file defines a number of parameters required for operation of wat_purge. A prototype of the wat_purge.cfg file can be retrieved from $NWISHOME/data/auxdata/wat_move wat_purge.cfg.master. This file is internally documented and contains a section to be edited by the user. The following is an example of a wat_purge.cfg file:

```
#! /bin/sh
#
#  wat_purge.cfg
#
#  Configuration file for wat_purge program
#
#  Notes:  The variable names must be in upper-case.
#          No space is allowed on either side of the equals sign.
#
#  The variables are defined as follows:
#    DBNUM           :  Database number for retrieval
#    START_DATE      :  Start date (yyyymmdd) for ADAPS UV deletions,
#                       NULL (‘’) => earliest date available
#    END_DATE        :  End date for UV deletions, NULL => current date
#    SITE_LIST_PATH  :  Path name of site-list file
#    FILE_LIST_PATH  :  Path name of file-list file
#    EDIT_LIST_PATH  :  Path name of edit-list file, NULL if not used
#
#  Modify the following lines as necessary:
#-----------------------------------------------------------------------------
DBNUM=01
START_YEAR=’’
END_YEAR=’’
SITE_LIST_PATH=./site_list
FILE_LIST_PATH=./file_list
EDIT_LIST_PATH=’’
```

A more detailed description of each of the variables follows:

DBNUM : 2-character database number.

START_DATE : Start date for deletion of ADAPS unit-values data. The date is in the form yyyymmdd. If a null (‘’) start date is specified, the data will be deleted from the earliest date available.

END_DATE : End date for ADAPS Unit-values deletions. If null, deletion will continue through the latest date available.

SITE_LIST_PATH : Path name for site-list file (section 7)

FILE_LIST_PATH : Path name for file-list file (section 8)

EDIT_LIST_PATH : Path name for edit-list file if user-defined editing is to be done (section 9).

6.3 Output from wat_purge

If the program was invoked using the wat_move main menu a log file named wat_purge.date.time.log will be created for each run of wat_purge. This file lists the NWIS files accessed with the number of records deleted as well as any error messages. This information will be provided to the user via e-mail if a cron entry is used to invoke the program (see section 3). The format is similar to that of the log file produced from wat_get described in section 4.3.

7. Description of the site-list file

The site-list file lists the sites for which data is to retrieved (wat_get) or deleted (wat_purge). The format of this file is:

Columns 1-5	:	Agency code (for example, “USGS ”)
Columns 6-20	:	Site number (for example, 05020500)

A partial site number, terminated with an asterisk, may be used. For example, the line:

```
USGS 0502*
```

would access all sites beginning with the characters “USGS 0502” while the line:

```
USGS*
```

would access all sites with agency code “USGS”.

The site-list file must be created by the user using any editor.

8. Description of file-list file

The file-list file indicates the NWIS files which are to be processed. A prototype of the file-list file may be obtained from $NWISHOME/data/auxdata/wat_move/wat_move.file_list.master. There are 21 records corresponding to the 21 NWIS files which may be processed with the wat_move software. Only column 1 is actually read by the software so columns 2-? may contain commentary. If column 1 contains a “Y” the file will be accessed. Otherwise it will not. The following is an example of a typical file-list file:

```
N  SITE  Site File
N  ADDC  ADAPS datum corrections file
N  ADDD  ADAPS data descriptor file
N  ADDV  ADAPS daily values file
N  ADIN  ADAPS instruments file
N  ADMS  ADAPS measurements file
N  ADPR  ADAPS processor file
N  ADRT  ADAPS ratings file
N  ADST  ADAPS shifts-by-time file
N  ADSV  ADAPS shifts-by-stage file
N  ADUV  ADAPS unit values file
N  GWCO  GWSI construction file
N  GWDI  GWSI discharge file
N  GWGE  GWSI geologic logs file
N  GWHY  GWSI hydraulics file
```

```
N  GWLE  GWSI water levels file
N  GWMI  GWSI miscellaneous file
N  GWOB  GWSI observation hdgs file
N  GWST  GWSI state water use file
Y  QW    QW File
N  ADSS  ADAPS statistics file
```

Note that in this example only the QW file is selected. Columns 3-6 in this example file-list file contain the NWIS file identification that is used to create transaction file names as defined in section 4.3. Note, also, that if QW data are to be stored using wat_put, a “qwindex” file is required. See section 11 for instructions for setting up the qwindex file.

9. NWIS File Edit Option

The wat_move software provides for limited editing of NWIS files. Edits are applied as follows:

- wat_get :
  Edit is applied after record is retrieved and before record is written to transaction file. Optionally, a record may be rejected for writing to the transaction file.
- wat_put :
  Edit is applied after transaction is read and before the record is stored in an NWIS file. Optionally, a transaction may be rejected for storing in the NWIS file.
- wat_purge : Edit is used to reject (by-pass) deletions.

Edit “commands”, if any, are specified in a file with path name defined by EDIT_LIST_PATH in the configuration file for the subsystem. Each line of the file contains one edit command. The general format of an edit command is:

action [defn] [if test] [connector test] [connector test] . . .

where:

“action” is the action to be taken. Possible values are:

“set” -- set sub-string in record to value

“reject” -- reject record for further processing

“defn” includes the location in the NWIS file record where data is to be stored and a definition of the data to be stored. defn is required for “set” action but is not used with “reject” actions. The general form of defn is:

(s1:s2) = value

where:

s1 : start location

s2 : end location

value : value to be stored, possible forms are:
‘aaaa’ : character string

(s1:s2) : substring in record location s1-s2 inclusive

"if" is required if any testing is to be done. Testing is required with "reject" actions.

"test" is a condition to be passed in order to perform "action". There may be more than one test with "and" and/or "or" connectors. Parentheses may not be used to group tests. For grouping, "and" connectors have higher priority than "or" connectors. For example:

A and B or C and D => (A and B) or (C and D)

The general form of a test is:

source relation value

where:

source indicates the value to be tested, possible forms are:
"node" : indicates test of source node (e.g., qvarsa)
"nwisid" : indicates test of NWIS file identification
(for example, "ADDC", see section 8.0).
(s1:s2) : indicates to test value in columns s1-s2 of NWIS record

relation is the test relation, possible values are:

```
=   !=   <   <=   >   >=   cn   nc
```

where "cn" denotes "contains" and "nc" denotes "does not contain"

value is the test value which is defined the same as "value" in "defn".

"connector" is the connector between tests. No connector is used with the first test but a connector is required after the first test. Possible values are "and" and "or".

The edit commands must contain at least one space between each of the components action, defn, and so forth. There can be no embedded spaces in a component.

For example:

```
"(1:5)" is OK, "(1 : 5)" is not.
```

The following are a few examples of using edit commands:

The command:

set (1:4) = 'WTMV' if nwisid != 'QW'

specifies that columns 1-4 are to be set to "WTMV" if the NWIS file identification is not "QW".

The command:

set (9:13) = 'WTMV' if nwisid = 'QW'

specifies that columns 9-13 are to be set to "WTMV" if the NWIS file identification is "QW".

The command:

reject if node = ‘qvarsa’ and nwisid = ‘ADDC’ and (1:5) = ‘MT009’

specifies that the record is to be rejected for processing if the source node is “qvarsa” and the NWIS file identification is “ADDC,” and if columns 1-5 of the NWIS record contain “MT009”.

10. Processing as a Cron Job

It may be desirable to run the wat_move software from the UNIX “cron” system. A crontab entry can be set up to invoke a cron job. A prototype of a shell script that could be executed as a cron job may be retrieved from $NWISHOME/data/auxdata/wat_move/wat_move.cronjob.master. The following is an example of a Bourne shell script for running wat_move from cron:

```
#! /sbin/sh
#  run wat_move as a cron job
#
#  define the bourne shell environment:
. /etc/profile
#  define the NWIS environment:
. /usr/opt/etc/nwis.profile
#  perform wrapper checks:
. $NWISHOME/util/nwo_set_nwis_env
#  include the next 2 statements only to change DB as defined in nwis.profile
NWISDB=nwisdb
export NWISDB
cd XXXXXX
$NWISHOME/bin/wat_move wat_get
```

Note that XXXXXX in the second to last line must be replaced with the pathname of the directory in which wat_move is to be executed. The subsystem must be specified as an argument to wat_move in the last line (wat_get in this example).

See your UNIX systems administrator for more information about cron processing.

11. Set Up qwindex File

When merging QW data from various databases, it is possible to have a conflict with the primary keys for the file. The wat_move software will handle the conflict but requires that a file, the qwindex file, be established. The following are instructions for setting up the qwindex file:

As user “nwis” ensure that the environment variables NWISDB and NWISHOME are correct. Then, enter the following 3 commands:

create_table $NWISHOME/support/qwindex

modify_table $NWISHOME/support/qwindex

cp $NWISHOME/data/auxdata/wat_move/qwindex.acl.master \
$NWISHOME/support/qwindex.acl

This setup will allow the users “nwis” and members of the UNIX group nw_dba to access the qwindex file.